ORACLE 11G RAC : PRE-REQUISITE VERIFICATIONS

Inderpal S. Johal, Data Softech Inc.

INTRODUCTION

This document can be used as reference in the future to install and configure the Real Application Cluster on Linux. It will cover all Pre-installation as well as Post-installation steps required for the Deployment of RAC in your environment. This document is written to provide help installing Oracle11*g* Real Application Clusters on x86 with Red Hat Linux 4. I will describe step-by-step architecture from installing Oracle Clusterware, Oracle11g Software and Database on ASM (Automatic Storage Management). I have tried to use all CLUVFY (Cluster Verification Utility) so as to make sure that all steps are completed successfully on all nodes.

ORACLE 11G RAC INSTALLATION OUTLINE

In this document, I will follow the below mentioned path to deploy 2-Node Oracle 11g RAC in Linux environment :

1. Perform Pre-Installation on Single Node
2. Install and configure ASMLib and Raw Devices on Single Node
3. Install Oracle 11g Clusterware on Single Node
4. Install Oracle 11g Software on Single Node
5. Configure Clustered Oracle ASM Instance on Single Node
6. Configure Non-RAC Instance on Single Node
7. Convert Non-RAC instance to RAC instance using the Same Clustered ASM Instance
8. Perform Pre-Installation on Second Node
9. Configure ASMLib and Raw Devices on Second Node
10. Add Oracle 11g Clusterware to Second Node
11. Add Oracle 11g Software to Second Node
12. Add ASM as well as Database instance to Second Node

At each Step of the deployment, I will provide the cluvfy command to validate the environment. In my configuration, I am using db02pn as First Node and db01pn as second Node.

PRE-INSTALLATION TASKS

You must perform the following step-by-step tasks for hardware, software and network requirements as well as for the Platform-specific pre-installation procedures. This will include OS patches for Cluster database as well as Kernel configuration. I will use Cluvfy (Cluster Verification Utility) at the end of each step so as to make sure that the task is completed successfully.

Memory

Oracle need a minimum of 1G physical memory on each cluster Node

```
[root@db02pn]# grep MemTotal /proc/meminfo
MemTotal:      16624992 kB
```

Swap Space

It should be 0.75 times of Physical RAM for RAM > 8G

```
[root@db02pn]#  grep SwapTotal /proc/meminfo
SwapTotal:      2096472 kB
```

As the Requirement is 0.75 * 32G and so we added more Swap as shown below

```
[root@db02pn]#  grep SwapTotal /proc/meminfo
[root@db02pn /]# dd if=/dev/zero of=/tempswap bs=1k count=9000000
9000000+0 records in
9000000+0 records out
```

Now we should change the file permissions:

```
[root@db02pn /]# chmod 600 /tempswap
```

Finally we format the "partition" as swap and add it to the swap space:

```
[root@db02pn /]# mkswap /tempswap
Setting up swapspace version 1, size = 9215995 kB
```

```
[root@db02pn /]# swapon /tempswap
```

It take some time and so have patience

```
[root@db02pn /]#  grep SwapTotal /proc/meminfo
SwapTotal:     11096464 kB
```

TMP Space

Verify the temp space in /tmp or add new tmp Mount point. OUI requires up to 400M of free space in /tmp directory

```
[root@db01pn]# df -h /tmp
Filesystem                              Size  Used Avail  Use%    Mounted on
/dev/mapper/VolGroup_ID_30884-LogVol3   4.0G   41M  3.7G   2%     /tmp
```

If for any reason, you do not have enough space in /tmp, you can temporarily create space in another file system and point your TEMP and TMPDIR to it for the duration of the install. Here are the steps to do this:

```
$ su -
# mkdir /mount_point/tmp
# chown root:root /mount_point/tmp
# chmod 1777 /mount_point/tmp
# exit
```

Login as Oracle

```
$ TEMP=/mount_point/tmp          # Used by Oracle
$ TMPDIR=/mount_point/tmp        # used by Linux Programs
$ export TEMP TMPDIR
```

ORACLE CLUSTERWARE AND DATABASE 11G SOFTWARE SPACE

4 Gigabytes of space to Install Oracle Software on each Node which includes around 1 G for Oracle Clusterware and around 3 G for Oracle Database Software

ORACLE_BASE=/home/oracle/app
ORACLE_HOME=$ORACLE_BASE/product/11.1/db_1
ORA_CRS_HOME=/home/oracle/crs/product/11.1/crs_1

Note : Please Note that Clusterware Home as shown above with ORA_CRS_HOME is not included in $ORACLE_BASE as ORA_CRS_HOME permission is changed to root after the Clusterware is completed and root.sh is executed on all nodes.

ORACLE DATABASE SPACE

Following is the space Requirement for Oracle Database related files. This will not include your application Schema space.

Oracle Shared Drive Configuration

File System Type	Partition	Size	Mount Point	ASM Diskgroup Name	File Types
2 Raw Device	/dev/sde1 /dev/sde2	500M	/dev/sde1 /dev/sde2		Oracle Cluster Registry (OCR) File Voting Disk
ASM	/dev/sde3	200G	/dev/sde3	+DG11G	Oracle Database Files
Total		**201G**			

SYSTEM ARCHITECTURE

Check the System architecture so as to verify that you have proper Oracle CD available for installation. The following command will display the processor type so as to ensure that you can install Oracle 64 bit software on the system

```
[root@db02pn]# grep “model name” /proc/cpuinfo
model name      : Intel(R) Xeon(TM) MP CPU 3.33GHz
model name      : Intel(R) Xeon(TM) MP CPU 3.33GHz
model name      : Intel(R) Xeon(TM) MP CPU 3.33GHz
model name      : Intel(R) Xeon(TM) MP CPU 3.33GHz
```

OPERATING SYSTEM USER AND GROUPS

Logon as Root User and execute the Following on First Node. This is the Group used for Installing Oracle Software

```
[root@db02pn]# /usr/sbin/groupadd –g 1000 dba
```

Logon as Root User and execute the Following on First Node. This group owns the Oracle Inventory, which is a catalog of all Oracle software installed on the system.

```
[root@db02pn]# /usr/sbin/groupadd -g 1001 oinstall
```

Logon as Root User and execute the Following on all First Node. This is the User who owns the Oracle Software

```
[root@db02pn]# /usr/sbin/useradd -u 1000 -g oinstall -G dba oracle
```

```
[oracle@db02pn ~]$ id oracle
uid=1000(oracle) gid=1001(oinstall) groups=1001(oinstall),1000(dba)
```

Set the Oracle Password

```
[root@db02pn]# passwd oracle
```

Verify and create the “nobody” user on all 2 nodes

```
[root@db02pn]# id nobody
uid=99(nobody) gid=99(nobody) groups=99(nobody)
```

If it does not exist, then create it with the following command

```
[root@db02pn]# /usr/sbin/useradd nobody
```

Network Requirements

- Each node must have at least two network adapters :
  1. Public network interface e.g eth0 as shown below and it has to be same on all RAC nodes later on.
  2. Private network interface (the interconnect) e.g eth2 as shown below and it has to be same on all RAC nodes later on.
- Check if you can ping all network address from each other node or run CLUVFY to check it for you.
- Use an IP address with an associated network name registered with DNS. If you don't have an available DNS, then record them with IP address in /etc/hosts file on all Nodes. So all Nodes have this information for all other nodes
- One Virtual IP(VIP) address must also be recorded in /etc/hosts file.
- A private IP address with a host name for each private network interface should also be recorded in /etc/hosts file.
- Use /sbin/ifconfig to see the interface name and associated IP address

```
[oracle@db02pn]$ cat /etc/hosts
172.XX.XX.34     db02    db02pn
172.XX.XX.42     db02-vip
10.0.0.2          db02-priv
[root@db02pn etc]# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:04:23:C9:08:22
          inet addr:172.30.106.34  Bcast:172.XX.XX.255  Mask:255.255.255.0
          inet6 addr: fe80::204:23ff:fec9:822/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:20102461 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11666099 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2920665269 (2.7 GiB)  TX bytes:2839439116 (2.6 GiB)
          Base address:0x6cc0 Memory:dece0000-ded00000

eth2      Link encap:Ethernet  HWaddr 00:13:72:40:89:70
          inet addr:10.0.0.2  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::213:72ff:fe40:8970/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:590 (590.0 b)
          Interrupt:209
```

NOTE: You do not have to configure the network alias names for the public VIP as it will be done by Oracle's Virtual Internet Protocol Configuration Assistant (VIPCA).

SOFTWARE REQUIREMENTS

SSH AND USER EQUIVALENCY

Follow the steps outlined below so as to allow ssh to talk to all nodes in the cluster as an **Oracle** user

Step 1. Logon as Oracle user

Step 2. Set up Directory on Both Node. Below is an example on Node1

```
[oracle@db02pn]$ rm –r ~/.ssh
[oracle@db02pn]$ mkdir ~/.ssh
[oracle@db02pn]$ chmod 700 ~/.ssh
```

Step 3. Create RSA and DSA Keys on Both Nodes. Below is an example on Node1

```
[oracle@db02pn]$ /usr/bin/ssh-keygen –t rsa
[oracle@db02pn]$ /usr/bin/ssh-keygen –t dsa
```

At the prompts:

Accept the default location for the key file

Enter and confirm a pass phrase that is different from the oracle user's password.

Now you will have 4 files created in ~/.ssh directory

Step 4. Add keys to an authorized Key file on Node 1

```
[oracle@db02pn]$ touch ~/.ssh/authorized_keys
[oracle@db02pn]$ cd ~/.ssh
```

Make sure that you have only two*.pub file.

```
[oracle@db02pn]$ ls –lt *.pub
```

Step 5. Create an initial temporary authorized file on each node suffixed with Node number as shown with Prompt

```
[oracle@db02pn]$ cat *.pub >auth_node2
[oracle@db01pn]$ cat *.pub >auth_node1
```

Step 6. Copy all of the auth_node1 on db02pn in ~/.ssh directory

```
[oracle@db02pn]$ cd ~/.ssh
[oracle@db02pn]$ scp oracle@db01pn:/home/oracle/.ssh/auth_node1
```

Step 7. Concatenate the contents of auth_node 2 into one Authorized file created in Step 4

```
[oracle@db02pn]$ cd ~/.ssh
[oracle@db02pn]$ cat auth_node* > authorized_keys
[oracle@db02pn]$ chmod 600 authorized_keys
```

Step 8. Copy the authorized_keys file to all other nodes

```
[oracle@db02pn]$ scp authorized_keys oracle@db01pn:~/.ssh
```

Step 9. Enabling SSH User Equivalency on all Cluster Node from where you want to invoke the OUI

```
[oracle@db02pn]$ exec /usr/bin/ssh-agent $SHELL
[oracle@db02pn]$ /usr/bin/ssh-add
```

Step 9. Verify the SSH user from all nodes to make sure that is not prompting for the Password.

```
[oracle@db02pn]$ ssh db01pn date
The authenticity of host 'db01pn (172.xx.xx.33)' can't be established.
RSA key fingerprint is 70:48:ed:95:1c:e4:3c:40:8e:16:66:be:17:26:44:91.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'db01pn,172.xx.xx.33' (RSA) to the list of known hosts.
Mon Jan  9 13:34:20 EST 2006
[oracle@db02pn]$ ssh db02pn date
```

SOFTWARE LIST FOR X86 64 BIT PLATFORM

KERNEL VERSION

The system must be running the following kernel version (or a later version).

Red Hat Enterprise Linux 4 (Update 3): 2.6.9-34.EL

To determine whether the required kernel errata is installed, use the following procedure for your Linux distribution:

```
[root@db01pn ~]# uname -r
2.6.9-55.ELsmp
```

Here *KERNEL VERSION* is 2.6 and Erata level is 55.EL

OPERATING SYSTEM X86 (64 BIT)

The system must be running one of the following

Red Hat Enterprise Linux AS/AS 4 (Update 3 or later)

To determine which distribution and version of Linux is installed, enter the following command:

```
# cat /etc/issue
```

```
[root@db01pn ~]# cat /etc/issue
Red Hat Enterprise Linux ES release 4 (Nahant Update 5)
Kernel \r on an \m
```

PACKAGE REQUIREMENTS

Red Hat Enterprise Linux 4 (update 3) need the following installed Packages (or later versions)

Required or Later version	Available Version
binutils-2.15.92.0.2-18	binutils-2.15.92.0.2-22
compat-db-4.1.25-9	compat-db-4.1.25-9
elfutils-libelf-0.97.5	elfutils-libelf-0.97.1-4
elfutils-libelf-devel-0.97-5	elfutils-libelf-devel-0.97.1-4
gcc-3.4.5-2	gcc-3.4.6-8
gcc-c++-3.4.5-2	gcc-c++-3.4.6-8
glibc-2.3.9.4-2.19	glibc-2.3.4-2.36
glibc-common-2.3.9.4-2.19	glibc-common-2.3.4-2.36
libaio-0.3.105-2	libaio-0.3.105-2
libaio-devel-0.3.105-2	libaio-devel-0.3.105-2
libstdc++-3.4.5-2	libstdc++-3.4.6-8
libstdc++-devel-3.4.5-2	libstdc++-devel-3.4.6-8
make-3.80-5	make-3.80-6.EL4
pdksh-5.2.14-30.3	pdksh-5.2.14-30.3
sysstat-5.0.5-7.rhel4	sysstat-5.0.5-15.0.1.el4

```
# rpm –qa |grep –i libaio   → to check if the package is installed
# rpm –Uvh *.rpm           → To Update the package
# rpm –ivh *.rpm          → To Insert the package
```

RED HAT 4: ORACLE AUTOMATIC STORAGE KERNEL DRIVER (ASMLIB)

See ASMlib Installation and Configuration later in the book

CVUQDISK Package Requirement for CLUVFY

- **Copy the cvuqdisk RPM package from media disk present in clusterware/rpm**
- **Login as root and execute the following to verify the package**

```
[root@db02pn]# rpm -qa|grep -i cvuqdisk
```

- **De-install the existing version if present**

```
# rpm -e cvuqdisk
```

- **Install the cvuqdisk RPM package**

```
[root@db02pn]# cd /mnt/clusterware/rpm
[root@db02pn]# ls –ltr cvuq*
-rw-r--r--  1 root root 5339 Oct 23 14:36 cvuqdisk-1.0.1-1.rpm

 [root@db02pn]# rpm -Uvh cvuqdisk-1.0.1-1.rpm
Preparing...          ########################################### [100%]
  1:cvuqdisk          ########################################### [100%]

[root@db02pn]# rpm -qa|grep -i cvuqdisk
cvuqdisk-1.0.1-1
```

Environmental Variable Setting

Step 1. Open the startup Shell script

Bash shell on Red Hat Enterprise Linux:

```
$ . ./.bash_profile
```

Step 2. Add the following to the above Startup shell script based on your Shell setting

```
umask 022
ORACLE_BASE=/home/oracle/app
ORACLE_HOME=$ORACLE_BASE/product/11.1/db_1
ORA_CRS_HOME=/home/oracle/crs/product/11.1/crs_1
LD_LIBRARY_PATH=$ORACLE_HOME/lib
TNS_ADMIN=$ORACLE_HOME/network/admin
PATH=$ORACLE_HOME/bin:$PATH
Export ORACLE_BASE ORACLE_HOME PATH LD_LIBRARY_PATH TNS_ADMIN
```

Oracle Software Home Directory

```
[root@db02pn]# mkdir -p /home/oracle/app/product/11.1/db_1
[root@db02pn]# chown -R oracle:oinstall /home/oracle/app/product/11.1/db_1
[root@db02pn]# chmod -R 775 /home/oracle/app/product/11.1/db_1
```

Oracle Clusterware Home Directory

```
[root@db02pn]# mkdir -p /home/oracle/crs/product/11.1/crs_1
[root@db02pn]# chown -R oracle:oinstall /home/oracle/crs/product/11.1/crs_1
[root@db02pn]# chmod -R 775 /home/oracle/crs/product/11.1/crs_1
```

Step 3. Add the following to the above Startup shell script(.bash_profile) to make sure sty interrupt is set properly to avoid OUI error.

```
if [ -t 0 ]; then
stty intr ^C
fi
```

Hangcheck-timer Module

The hangcheck-timer module uses a kernel-based timer that periodically checks the system task scheduler to catch delays in order to determine the health of the system. If the system hangs or pauses, the timer resets the node.

This module is now included with Red Hat Linux AS starting with kernel versions 2.4.9-e.12 and higher. Use the following to ensure that you have the module included:

```
[oracle@db01pn ~]$ find /lib/modules -name "hangcheck-timer.ko"
/lib/modules/2.6.9-5.ELsmp/kernel/drivers/char/hangcheck-timer.ko
/lib/modules/2.6.9-55.EL/kernel/drivers/char/hangcheck-timer.ko
/lib/modules/2.6.9-55.ELsmp/kernel/drivers/char/hangcheck-timer.ko
/lib/modules/2.6.9-5.EL/kernel/drivers/char/hangcheck-timer.ko
```

Make sure hangcheck-timer module is running on all nodes

```
[root@db02pn]# /sbin/lsmod |grep –i hang
```

If the hangcheck-timer module is not listed for any node, enter a command similar to the following to start the module on that node:

```
[root@db02pn]# /sbin/insmod hangcheck-timer hangcheck_tick=30 hangcheck_margin=180
```

To ensure that the module is loaded every time the system restarts, verify that the local system startup file contains the command shown in the previous step, or add it if necessary:

/etc/rc.d/rc.local file.

These values need to be available after each reboot of the Linux server. To do this, make an entry with the correct values to the /etc/modprobe.conf file as follows:

```
[root@db02pn]#  echo "options hangcheck-timer hangcheck_tick=30 hangcheck_margin=180" >>
/etc/modprobe.conf
```

To manually load the hangcheck-timer kernel module and verify it is using the correct values defined in the /etc/modprobe.conf file, run the following command:

```
[root@db02pn]# modprobe hangcheck-timer
[root@db02pn]# grep Hangcheck /var/log/messages | tail -2
Jul 23 12:29:32 db02pn kernel: Hangcheck: starting hangcheck timer 0.9.0 (tick is 30 seconds, margin is 180 seconds).
Jul 23 12:29:32 db02pn kernel: Hangcheck: Using monotonic_clock().
```

Setting Shell limits for Oracle User

To increase the shell limits:

1. Add the following lines to the /etc/security/limits.conf file:

```
oracle soft nproc 2047
oracle hard nproc 16384
oracle soft nofile 1024
oracle hard nofile 65536
```

2. Add or edit the following line in the /etc/pam.d/login file, if it does not already exist:

```
session required pam_limits.so
```

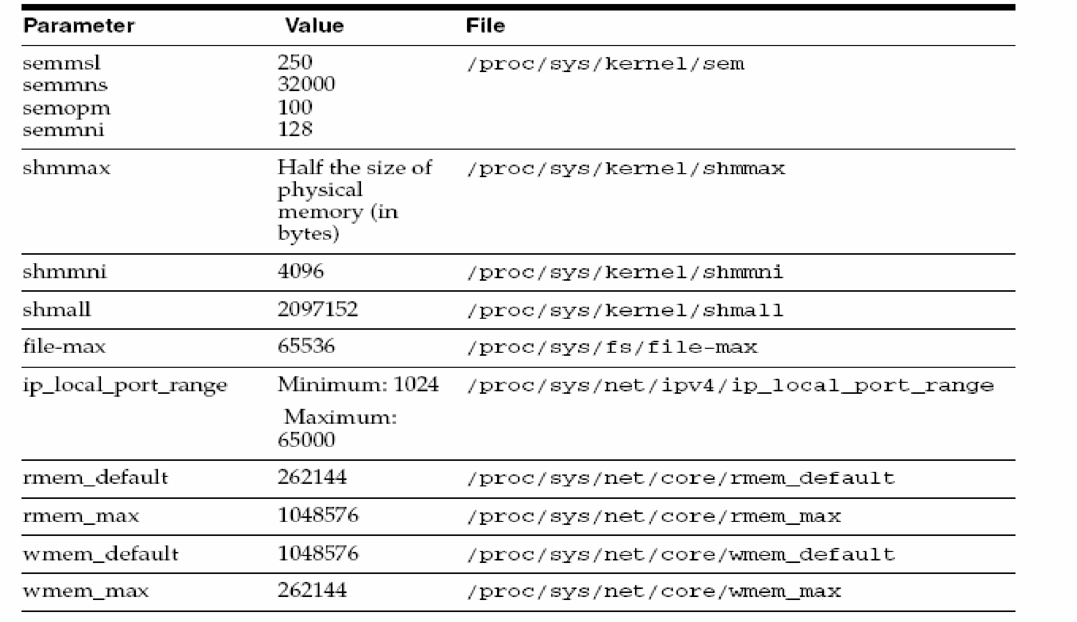
Node Time Requirements

Make sure that all Node of the cluster have the same or very close system time. Use the following command to change the date on any system

```
[root@db02pn ~]# date mmddHHMI
```

Kernel Parameters

Kernel Recommended Values are



Parameter	Value	File
semmsl semmns semopm semmni	250 32000 100 128	/proc/sys/kernel/sem
shmmax	Half the size of physical memory (in bytes)	/proc/sys/kernel/shmmax
shmmni	4096	/proc/sys/kernel/shmmni
shmall	2097152	/proc/sys/kernel/shmall
file-max	65536	/proc/sys/fs/file-max
ip_local_port_range	Minimum: 1024 Maximum: 65000	/proc/sys/net/ipv4/ip_local_port_range
rmem_default	262144	/proc/sys/net/core/rmem_default
rmem_max	1048576	/proc/sys/net/core/rmem_max
wmem_default	1048576	/proc/sys/net/core/wmem_default
wmem_max	262144	/proc/sys/net/core/wmem_max

Make sure RAC node name is not listed in loopback address

Ensure that the node names (linux1 or linux2) are **not** included for the loopback address in the /etc/hosts file. If the machine name is listed in the in the loopback address entry as below:

```
127.0.0.1      linux1 localhost.localdomain localhost
```

it will need to be removed as shown below:

```
127.0.0.1      localhost.localdomain localhost
```

FINAL CLUVY TEST

CHECKING THE SYSTEM SETUP BEFORE PROCEEDING FOR CLUSTERWARE INSTALLATION

```
          $ ./runcluvfy.sh comp sys -p crs -r 11gR1 -n db02pn
          Verifying system requirement
          Checking system requirements for 'crs'...
          …
          …
          System requirement passed for 'crs'
      Verification of system requirement was successful.
```

```
          $ ./runcluvfy.sh comp sys -p crs -r 11gR1 -n db02pn -verbose
          Verifying system requirement
          Checking system requirements for 'crs'...
          Check: Total memory
            Node Name     Available                 Required                  Comment
            ------------  ------------------------  ------------------------  ----------
            db02pn  15.86GB (16627268KB)      1GB (1048576KB)           passed
          Result: Total memory check passed.
          …
          …
          System requirement passed for 'crs'
          Verification of system requirement was successful.
```

DISCOVER ALL SHARED STORAGE

```
 $  ./runcluvfy.sh comp ssa -n db02pn -s /dev/sde1 –verbose   -----→ Used for OCR file
       Verifying shared storage accessibility
       Checking shared storage accessibility...
       "/dev/sde1" is shared.
       Shared storage check was successful on nodes "db02pn".
       Verification of shared storage accessibility was successful.
```

```
 $  ./runcluvfy.sh comp ssa -n db02pn -s /dev/sde2 –verbose    -------→ Used for Voting Disk
       Verifying shared storage accessibility
       Checking shared storage accessibility...
       "/dev/sde2" is shared.
       Shared storage check was successful on nodes "db02pn".
       Verification of shared storage accessibility was successful.
```

```
 $ ./runcluvfy.sh comp ssa –n db02pn -s /dev/sde3 -verbose   -------→ Used for ASM based Database files
       Verifying shared storage accessibility
       Checking shared storage accessibility...
       "/dev/sde3" is shared.
       Shared storage check was successful on nodes "db02pn".
       Verification of shared storage accessibility was successful.
```

VERIFY HARDWARE AND OPERATING SYSTEM SETUP:

```
$ ./runcluvfy.sh stage -post hwos -n db02pn
Performing post-checks for hardware and operating system setup

Checking node reachability...
Node reachability check passed from node "db02pn".
…
…
Post-check for hardware and operating system setup was successful.
```

VERIFY CONNECTIVITY BETWEEN THE CLUSTER NODES THROUGH ALL OF THE AVAILABLE NETWORK INTERFACES

```
$ ./runcluvfy.sh comp nodecon -n db02pn -i eth2 -verbose
Verifying node connectivity
Checking node connectivity...
…
…
Check: Node connectivity for interface "eth2"
Result: Node connectivity check passed for interface "eth2".
Result: Node connectivity check passed.
Verification of node connectivity was successful.

$ ./runcluvfy.sh comp nodecon -n db02pn -i eth0 -verbose
Verifying node connectivity
Checking node connectivity...
…
…
Check: Node connectivity for interface "eth0"
Result: Node connectivity check passed for interface "eth0".
Result: Node connectivity check passed.
Verification of node connectivity was successful.
```

VERIFY USER EQUIVALENCE FOR NODES.

It will first check ssh and if it fails, then check with rsh

```
$ ./runcluvfy.sh comp admprv -n db02pn -o user_equiv -verbose
 Verifying administrative privileges
 Checking user equivalence...
 Check: User equivalence for user "oracle"
  Node Name                              Comment
  ------------------------------------  ------------------------
  db02pn                         passed
Result: User equivalence check passed for user "oracle".
Verification of administrative privileges was successful.
```